计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(9):265−271 [doi: 10.15888/j.cnki.csa.009221]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

句法依存引导的自注意力机制的中文分词①

周保途

(科大国创云网科技有限公司, 合肥 230088)
通信作者: 周保途, E-mail: zhoubaotu1323@163.com

摘　要: 在前人工作的基础上, 提出句法依存引导的自注意力机制以融合句法依存知识去提升中文分词的性能, 使得自注意力机制只关注那些对当前字符的分词标签有句法依存影响的字符, 学习它们对于当前字符的影响程度, 另外, 该文根据句法依存树对引导后的自注意力机制进行位置编码. 实验结果表明, 模型相较于 baseline 取得了性能上的提升, 同时模型对未登录词的识别能力也有所提升.

关键词: 句法依存; 位置编码; 自注意力机制; 中文分词

引用格式: 周保途.句法依存引导的自注意力机制的中文分词.计算机系统应用,2023,32(9):265–271. http://www.c-s-a.org.cn/1003-3254/9221.html

Chinese Word Segmentation of Self-attention Mechanisms Guided by Syntactic Dependence

ZHOU Bao-Tu

(GuoChuang Cloud Technology Co. Ltd., Hefei 230088, China)

Abstract: Based on previous work, this study proposes that the self-attention mechanism guided by syntactic dependency can integrate syntactic dependency knowledge to improve the performance of Chinese word segmentation so that the self-attention mechanism can only focus on those characters that have syntactic dependency influence on the current character's word segmentation label and learn their influence degree on the current character. In addition, this study performs positional encoding on the self-attention mechanism guided by syntactic dependency trees. The experimental results show that the model has improved its performance compared with the baseline, and the recognition ability of the model for unregistered words has been strengthened.

Key words: syntactic dependence; positional encoding; self-attention mechanisms; Chinese word segmentation

中文分词为自然语言处理 (natural language processing, NLP) 的基础性任务之一, 对于各种下游任务有直接影响. 总结以往的中文分词研究, 可以分为以下几种方案: 1) 基于词典的分词方法[1]; 2) 构建分词规则的方法[2]; 3) 依据机器学习模型的方法[3,4]; 4) 依据深度学习模型的方法[5,6]. 将中文分词建模为序列标注任务是分词的主流方法之一[7,8], 该方法给每一个字符赋予对应的分词标签以明确某个字符在词中的具体位置来完成分词, 当前最主流的分词标签集合为: {S: 单字成词, B: 词的起始位置, M: 词的中间位置, E: 词的结束位置}.

随着深度学习技术和计算机硬件的不断发展, 构建并训练拥有大规模参数的预训练语言模型 (例如: BERT[9]、XLNet[10]、ZEN[11] 等) 成为可能, 预训练语言模型提升了大部分自然语言处理任务的性能上限, 中文分词自然也不例外. 近年来, 关于融合句法知识的深度学习模型的研究有了一定发展. 其中, Tian 等人[12]提出一种融合句法知识的双通道自注意力模型, 提高了分词和词性标注的联合学习模型的性能. 韩虎等人[13]构建一种融合句法知识和位置信息的交互图自注意力

① 收稿时间: 2023-02-20; 修改时间: 2023-03-22; 采用时间: 2023-03-30; csa 在线出版时间: 2023-06-09
CNKI 网络首发时间: 2023-06-12

机制模型, 提升了方面级情感分析的识别能力. Chen 等人[14] 利用键值记忆网络融合句法知识, 对实体内部单词级别的关系和跨实体间关系进行建模, 构建关系抽取模型, 取得了性能上的提升. Zhang 等人[15] 通过可视化发现, 在阅读理解任务中, 自注意力机制会关注一些不太重要的虚词, 所以利用句法依存知识对自注意力机制加以约束.

以上的研究结果表明, 将句法知识融入到 NLP 深度学习模型中会有更好的表现, 因此, 有理由相信, 融合句法知识的中文分词模型能够进一步提升模型分词的能力. 本文在 Zhang 等人[15] 研究基础上, 将句法依存中词之间的依存关系, 转化为字符之间的关系, 并根据句法依存树将位置信息编码到模型中, 构建一种句法依存引导的自注意力机制的中文分词模型 (Chinese word segmentation model based on self-attention mechanism guided by syntactic dependency, AM-GSD).

1 方 法

传统的自注意力机制对所有字符不加显示约束地计算注意力向量, 这会使得自注意力机制变得分散, 使分词模型更容易受到一些不必要信息的干扰. 本文的出发点是利用句法依存关系对自注意力机制加以引导, 从而削减一些噪声, 进一步提升模型分词的性能.

1.1 算法框架概述

本文提出的中文分词模型 AM-GSD 的结构如图 1 所示.

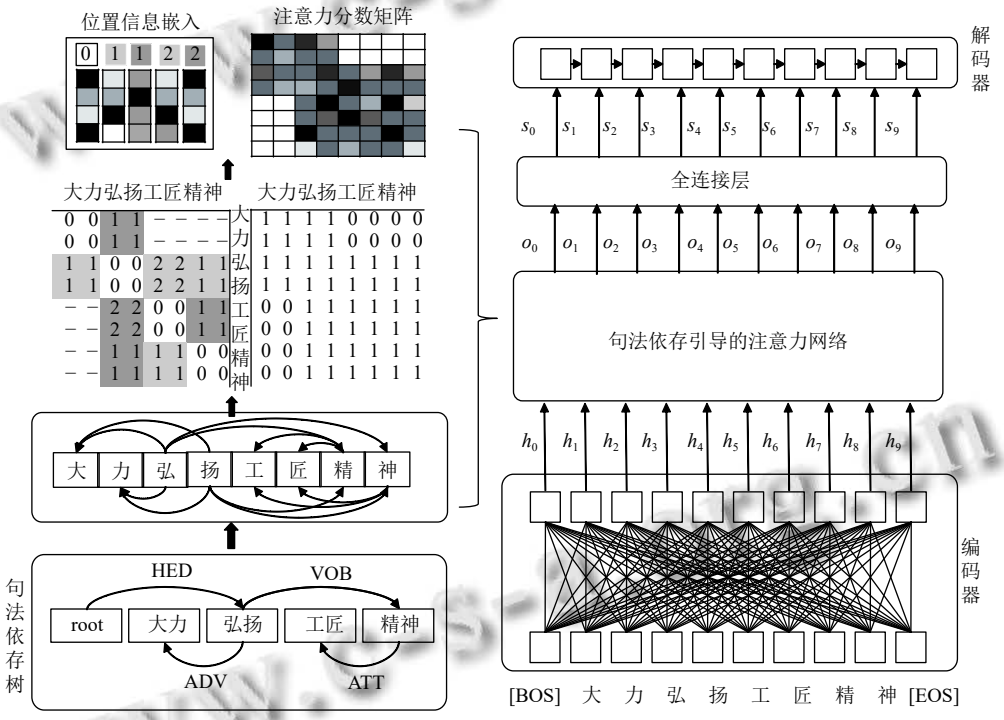

图 1 AM-GSD 模型结构

AM-GSD 通过句法依存树对自注意力机制进行引导, 把句法知识融入到模型中, 使得模型着重关注那些对当前字符的分词标签有句法依存影响的字符, 学习它们对于当前字符的影响程度. 同时, AM-GSD 基于句法依存树进行位置编码, 把句法依存树上的位置信息融入到模型中. AM-GSD 首先利用编码器将输入的文本序列转化为相对应的语义表征向量, 然后利用句法依存工具获取到相对应的句法依存树, 并利用句法依存树引导自注意力机制, 将语义表征向量和句法依存树上的位置信息输入到引导后的自注意力网络中, 最后将输出的语义表征向量输入到解码器中得到相应的分词标签. 下面将分别对句法依存引导的自注意力机制, 编码器-解码器和损失函数进行详细的介绍.

1.2 句法依存引导的自注意力机制

句法依存树刻画的是文本序列中各个单词之间的依存关系. 如图 2 所示, 词汇“弘扬”和词汇“精神”存在着这种依存关系, 其中, 词汇“弘扬”是支配词, 词汇“精神”是从属词, 二者被依存弧所连接, 依存弧上的标识

为依存关系类型. 每一个句子都会有一个依存关系核心词, 其与句子中的其他词汇存在着直接或间接的联系.

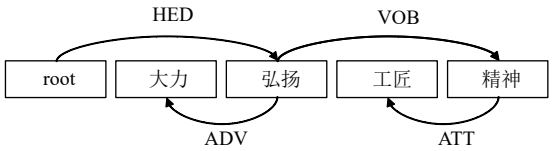

图 2 句法依存树

如图 3 所示, 根据句法依存树, 我们可以将词与词之间的依赖关系转化为字符与字符之间的关系, 从而可以对自注意力机制进行引导, 把句法知识融入到分词模型中.

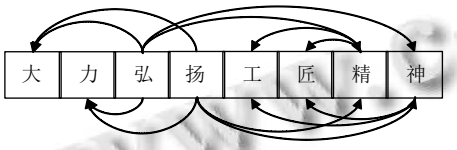

图 3 字符之间的关系

假设存在文本序列X, 其长度为n, 文本序列中的字符为x, 文本对应的分词标注序列为Y, 字符x_i所对应的正反向连通节点集为S_i, 由此构造关系矩阵M, 其中:

$$M_{ij}=\begin{cases}1, & i=j \text{ or } c_j\in S_i \text{ or } S_i=S_j\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

如图 3 所示, “大”的反向连通节点为“弘”和“扬”, 反之, “弘”和“扬”的正向连通节点集中也有“大”, 因此, 示例句子对应的矩阵的第 1 行中 (1, 1)(1, 3)(1, 4) 位置上的数值为 1, 第 1 列 (3, 1)(4, 1) 位置上的数值为 1.

本文将编码器 BERT 的输出作为句法依存引导的自注意力机制的输入, 句法依存引导的自注意力机制的前向传播计算过程如式 (2)–式 (4) 所示:

$$o_i=\sum_{j=1}^{n}a_{ij}(h_jW^v) \tag{2}$$

$$a_{ij}=ReLU(e_{ij}) \tag{3}$$

$$e_{ij}=M_{ij}\left(\frac{(h_iW^q)(h_jW^k)^{\mathrm{T}}}{\sqrt{d}}\right) \tag{4}$$

其中, W^q, W^k和W^v是可训练的参数矩阵, d为嵌入维度. 计算得到的结果会被输入到一个全连接层中得到非规范化的概率, 最后将非规范化的概率输入到解码器中得到分词标签.

位置编码 (positional encoding, PE) 在自注意力机制中的作用十分重要, 其可以将文本序列中单词的相对位置信息融入到自注意力机制中, 从而可以避免只传递词嵌入所产生的混乱问题. 基于此, 本文受前人工作[16,17] 的启发, 根据句法依存关系获得位置编码信息.

对于一个文本序列X, x_i和x_j表示序列中的两个字符, 构造位置关系矩阵P, 其中:

$$P_{ij}=\begin{cases}l_{ij}, & x_i\text{支配}x_j\\ \neg l_{ij}, & x_j\text{支配}x_i\\ 0, & x_i\text{和}x_j\text{共词}\\ -1, & x_i\text{和}x_j\text{无关系}\end{cases} \tag{5}$$

其中, l_{ij}表示正向跳数, $\neg l_{ij}$表示反向跳数. 具体的, 如图 4 所示, “弘”是“大”的反向 1 跳节点, 用图 4 中的深灰色填充的数字 1 来表示; “工”是“弘”的正向 2 跳节点, 用图 4 中浅灰色填充的数字 2 来表示; “大”分别是“大”和“力”的 0 跳, 用图 4 中的白色填充的数字 0 来表示, 图 4 中白色填充的符号“–”代表两个字符之间没有位置关系.

大	力	弘	扬	工	匠	精	神	
0	0	1	1	–	–	–	–	大
0	0	1	1	–	–	–	–	力
1	1	0	0	2	2	1	1	弘
1	1	0	0	2	2	1	1	扬
–	–	2	2	0	0	1	1	工
–	–	2	2	0	0	1	1	匠
–	–	1	1	1	1	0	0	精
–	–	1	1	1	1	0	0	神

图 4 位置关系矩阵

在算法设计中, 可以初始化两个位置信息矩阵$T^k\in R^{(2l_{\max}+1)\times d}$和$T^v\in R^{(2l_{\max}+1)\times d}$, 从而根据$P_{ij}$索引出$x_i$和$x_j$之间的句法依存位置信息向量$T^v_{ij}\in R^d$和$T^k_{ij}\in R^d$. 于是, 加入句法依存关系位置信息后的自注意力机制的前向传播计算过程如式 (6)–式 (8) 所示:

$$o_i=\sum_{j=1}^{n}a_{ij}(h_jW^v+T^v_{ij}) \tag{6}$$

$$a_{ij}=ReLU(e_{ij}) \tag{7}$$

$$e_{ij}=M_{ij}\left(\frac{(h_iW^q)(h_jW^k+T^k_{ij})^{\mathrm{T}}}{\sqrt{d}}\right) \tag{8}$$

其中, $l_{\max}$表示语料集中字符x_i和字符x_j的最大正向跳数.

1.3 编码器-解码器

(1) 编码器

在文本序列X输入到 AM-GSD 网络之前, 先通过

编码器将字符x_i编码成一个包含文本语义信息的表征向量h_i, 如式 (9) 所示:

$$h_i = Encoder(x_i) \tag{9}$$

其中, $Encoder(\cdot)$表示编码器, 本文选用 BERT 作为编码器, 具体所使用的模型为 BERT-base-chinese.

(2) 解码器

将h_i输入到 AM-GSD 中获得表征向量o_i, 然后利用一个全连接层将其投射到概率空间上获得非规范化的概率s_i, 最后经过解码器获得相应的预测分词标签, 如式 (10) 所示:

$$\hat{y}_i = Decoder(s_i) \tag{10}$$

其中, $Decoder(\cdot)$表示解码器, 本文选用 CRF 作为解码器.

1.4 损失函数

本文采用负对数似然损失函数的标准反向传播算法进行训练, 计算方法如式 (11) 所示:

$$L(\theta) = -\sum_i \log(p(y_i|x_i;\theta)) \tag{11}$$

其中, θ为模型参数.

2 实验

2.1 数据集选取

本文选取中文分词常用的 PKU, MSR, AS 和 CITYU 这 4 个数据集作为本文的评测数据集, 并按照官方标准划分训练集和测试集, 上面的 4 个数据集中, AS 和 CITYU 是繁体中文数据集, 需将其转化为简体中文后用于模型的评测. 4 个数据集的各项统计指标如表 1 所示.

表 1 数据集各项指标

数据集	指标	字数	词数	字符类	词类	OOV
PKU	训练	1 826k	1 110k	5k	55k	—
	测试	173k	104k	3k	13k	5.8
MSR	训练	4 050k	2 368k	5k	88k	—
	测试	184k	107k	3k	13k	2.7
AS	训练	8 368k	5 500k	6k	141k	—
	测试	198k	123k	4k	19k	4.3
CITYU	训练	2 403k	1 456k	5k	69k	—
	测试	68k	41k	3k	9k	7.2

2.2 评测指标

本文所选评测指标为F_1得分和R_{oov}, 具体的计算方法如式 (12)–式 (13) 所示:

$$F_1 = \frac{2\times 准确率 \times 召回率}{准确率 + 召回率} \tag{12}$$

$$R_{\text{oov}} = \frac{正确分类OOV词的数量}{测试集中OOV词的数量}\times 100\% \tag{13}$$

其中, F_1得分用于评测中文分词的性能; R_{oov}则用于评测分词模型的泛化能力.

2.3 实验环境

本文实验的编程环境为 Python 3.7.12, 深度学习框架 PyTorch 的版本为 1.12.1+cu113, 实验的运行环境为 Ubuntu 18.04.5; CPU 型号为 Intel(R) Xeon(R) Gold 6330 CPU @ 2.00 GHz, GPU 为 Tesla A100, 显存为 40 GB, CUDA 的版本为 11.6.

2.4 参数设置

本文实验的一些重要参数设置如表 2 所示.

表 2 一些重要参数

参数	参数值
位置信息嵌入	768
BERT嵌入向量维度	768
初始学习率	5E–6
max_seq_length	300
最大迭代轮数	100
batch_size	8

2.5 算法伪代码

为了更直观的理解本文算法过程, 将算法伪代码整理如算法 1.

算法 1. 句法依存引导的自注意力机制的分词模型

输入: 带有标注的数据 D; 最大迭代轮数 E.
输出: 在测试集上损失最小的模型model*.

初始化: 模型参数θ包括位置嵌入矩阵E, 参数矩阵W^q, W^k和W^v.
for i=1 to E do
  for $Batch \in D$ do
    for 文本序列X, 文本序列标注Y, 关系矩阵M
      位置嵌入索引矩阵IDX in $Batch$ do
      for $x_i \in X$ do
        $h_i \leftarrow Encoder(x_i)$;
        $T^v_{ij} \leftarrow E(IDX_{ij})$;
        $T^k_{ij} \leftarrow E(IDX_{ij})$;
        $e_{ij} \leftarrow M_{ij}\left(\frac{(h_iW^q)(h_jW^k+T^k_{ij})}{\sqrt{d}}\right)^{\mathrm{T}}$;
        $a_{ij} \leftarrow ReLU(e_{ij})$;
        $o_i \leftarrow \sum_{i=1}^{n} a_{ij}(h_jW^v+T^v_{ij})$;
      end
      $O \leftarrow [o_1,\cdots,o_n]$;
      $S \leftarrow Dense(O)$;
      $\hat{Y} \leftarrow Decoder(S)$;
    end
    $loss \leftarrow L(Y_{\text{batch}},\hat{Y}_{\text{batch}})$;
    $update\ \theta$;

$\hat{Y}_{test}$←model(X_{test});
$loss_{test}$←$L(Y_{test},\hat{Y}_{test})$;
if $loss_{test}$<$loss_{testmin}$
model*←model;
$loss_{testmin}$←$loss_{test}$
end
renturn model*

2.6 对比实验

为了评测 AM-GSD 模型的有效性, 选取近几年来的分词模型与其进行对比, 对比实验结果如表 3 所示.

表 3 对比实验结果 1 (%)

模型	PKU		MSR		AS		CITYU	
	F_1	R_{oov}	F_1	R_{oov}	F_1	R_{oov}	F_1	R_{oov}
文献[18]	96.56	85.53	98.31	85.32	96.62	79.63	97.74	87.45
文献[19]	94.41	78.91	98.05	78.92	96.44	76.39	96.61	86.91
文献[20]	94.32	72.64	96.04	71.60	94.75	75.37	95.55	81.40
文献[21]	96.15	69.88	97.78	64.20	95.22	77.33	96.22	73.58
文献[22]	96.64	**87.12**	97.80	**86.78**	96.02	79.10	97.02	84.24
文献[23]	96.32	—	98.19	—	**97.43**	—	97.80	—
文献[24]	**96.7**	81.6	97.9	84.0	96.7	78.3	97.6	**90.1**
AM-GSD	96.6	86.88	**98.45**	86.67	96.85	**81.82**	**97.86**	88.09

从表 3 可以看到, 文献 [18] 将文本序列中的成词信息融入到分词模型中; 文献 [19] 将 Transformer 和 CRF 结合起来构建分词模型; 文献 [20] 构建了一种多准则的分词模型; 文献 [21] 将 n-gram 片段语义信息融入到 LSTM 分词模型中; 文献 [22] 提出了一种基于机器阅读理解模型的分词方法, 针对性地解决中文分词序列标注模型很难获取句子的长距离语义依赖的问题; 文献 [23] 首先训练一个分词模型来生成初始的分词结果, 然后用语言模型的预测结果来评价分词结果的质量, 最后用增强的最小风险训练方法来提高分词模型的性能; 文献 [24] 是一种多标准分词方法, 其设置一个私有层和一个公共层来分别学习每个标准的特定特征和所有标准的公有特征, 然后将其二者拼接, 输入到 CRF 中, 计算得分函数. 以上的研究成果都取得了不错的效果, 可以看到本文提出的模型在其中的表现还是不错的, 在 3 个指标上取得了最好的成绩.

上面是从绝对角度来对比的, 下面本文将与另外一些中文分词 SOTA 模型进行提升量的对比, 对比结果如表 4 所示.

模型 1 利用键值记忆网络将成词信息注入到模型中; 模型 2 构建了一种带标签注意力的记忆网络, 将标签的依赖关系融入到模型中. 这两个模型都取得了不错的提升效果, 从表中可以看出本文的模型在提升量上的对比上并不处于优势地位, 但还是有一定的提升, 而且本文 baseline 模型的指标数值较高, 说明 AM-GSD 在一个较高的基础水平上仍有一定的提升.

表 4 对比实验结果 2 (%)

模型	PKU		MSR		AS		CITYU	
	F_1	R_{oov}	F_1	R_{oov}	F_1	R_{oov}	F_1	R_{oov}
模型1[6] (Baseline)	96.32	85.04	97.98	85.52	96.34	77.75	97.63	86.66
模型1	96.51	86.76	98.28	84.44	96.58	78.48	97.80	87.57
模型1 (提升量)	**0.19**	**1.72**	**0.30**	**−1.08**	**0.24**	**0.73**	**0.17**	**0.91**
模型2[25] (Baseline)	96.51	86.76	98.28	84.44	96.58	78.48	97.80	87.57
模型2	96.70	87.03	98.35	86.52	97.04	83.21	97.88	87.68
模型2 (提升量)	**0.19**	**0.27**	**0.07**	**2.08**	**0.46**	**4.73**	**0.08**	**0.11**
AM-GSD (Baseline)	96.53	85.80	98.28	86.00	96.70	81.32	97.76	86.73
AM-GSD	96.60	86.88	98.45	86.67	96.85	81.82	97.86	88.09
AM-GSD (提升量)	**0.07**	**1.08**	**0.17**	**0.67**	**0.15**	**0.50**	**0.10**	**1.36**

2.7 消融实验

为了进一步验证 AM-GSD 模型的有效性, 对其进行消融实验, 实验的结果如表 5 所示.

表 5 消融实验结果 (%)

模型	PKU		MSR		AS		CITYU	
	F_1	R_{oov}	F_1	R_{oov}	F_1	R_{oov}	F_1	R_{oov}
BERT-crf	96.53	85.80	98.28	86.00	96.70	81.32	97.76	86.73
AM-GSD	96.56	86.44	98.44	86.36	96.76	80.81	97.82	86.86
AM-GSD (PE)	**96.60**	**86.88**	**98.45**	**86.67**	**96.85**	**81.82**	**97.86**	**88.09**

本文有两个改进之处.

(1) 利用句法依存树引导自注意力机制, 使模型只关注那些对当前字符的分词标签有句法依存影响的字符, 将句法知识融入到模型之中.

(2) 针对引导后的自注意力机制提出一种基于句法依存树的位置编码方式, 对引导后的自注意力机制进行位置编码.

从 BERT-crf 和 AM-GSD 的对比结果来看, AM-GSD 相较于 BERT-crf 在 PKU、MSR、AS 和 CITYU 的F_1得分都有提升, 分别为 0.03%、0.16%、0.06% 和 0.06%, 模型的R_{oov}除了 AS 数据集之外, 分别提升 0.64%、0.36% 和 0.13%. 对比结果说明, 利用句法依存树对自注意力机制进行引导, 将句法知识融入的分词

模型中, 能够提升模型分词的性能和泛化能力.

从 AM-GSD 和 AM-GSD (PE) 的对比结果来看, AM-GSD(PE) 相较于 AM-GSD 在 PKU、MSR、AS 和 CITYU 的F_1得分都有提升, 分别为 0.04%、0.01%、0.09% 和 0.04%, 模型的R_{oov}分别提升 0.44%、0.31%、1.01% 和 1.23%, 其中, 引入位置信息后的模型在 AS 数据集上相较于 BERT-crf 提升 0.5%. 对比结果说明了针对引导后的自注意力机制提出的这种基于句法依存树的位置编码方式能够提升模型的分词性能和泛化能力.

2.8 样例分析

为了更直观的说明 AM-GSD 的作用, 本文以文本序列“大力弘扬工匠精神”为例, 画出其热力图如图 5 和图 6 所示.

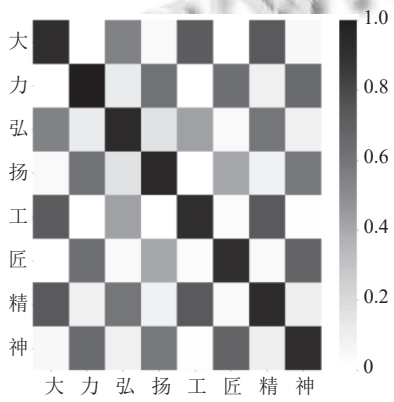

图 5 未加句法引导的自注意力机制可视化结果

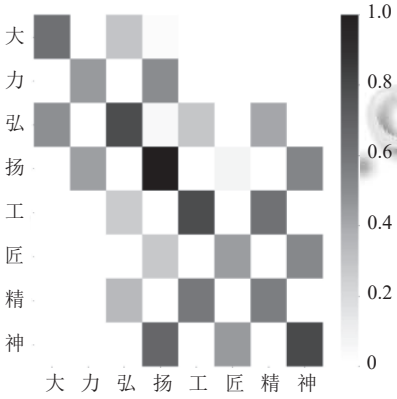

图 6 句法引导的自注意力机制可视化结果

通过图 5 和图 6 对比得出, BERT-crf (图 5) 的自注意力比较分散, 每个字符之间的相关性比较复杂; 而 AM-GSD (图 6) 的自注意力比较集中, 字符之间的相关性在句法依存关系的引导下得到了一定程度的简化. 这也印证了, AM-GSD 可以很好地基于句法依存关系对自注意力机制进行引导, 相当于对分词模型进行剪枝, 从而进一步提升了模型的分词性能.

3 结论与展望

本文提出一种句法依存引导的自注意力机制分词模型, 并针对性地提出一种基于句法依存树的位置编码方式. 模型将句法依存知识融入到分词模型中, 提高了模型分词的性能和泛化能力.

本文并未考虑句法依存关系类型, 获得的句法依存关系也存在一定的错误, 这会对分词模型产生负面影响. 在未来的研究工作中, 可以将引入的句法依存知识进一步完整化.

参考文献

1 Zhang Q, Liu XY, Fu JL. Neural networks incorporating dictionaries for Chinese word segmentation. Proceedings of the 2018 AAAI Conference on Artificial Intelligence. New Orleans: AAAI, 2018. 5682–5689. [doi: 10.1609/aaai.v32i1.11959]

2 Wu AD, Jiang ZX. Word segmentation in sentence analysis. Proceedings of the 1998 International Conference on Chinese Information Processing. Beijing: Tsinghua University Press, 1998. 169–180.

3 Asahara M, Goh CL, Wang XJ, *et al*. Combining segmenter and chunker for Chinese word segmentation. Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing. Sapporo: ACM, 2003. 144–147. [doi: 10.3115/1119250.1119270]

4 Fan C, Li Y. Research on Chinese word segmentation based on conditional random fields. Proceedings of the 17th International Conference on Intelligent Computing Theories and Application. Shenzhen: Springer, 2021. 316–326. [doi: 10.1007/978-3-030-84529-2_27]

5 Qun N, Yan H, Qiu XP, *et al*. Chinese word segmentation via BILSTM+Semi-CRF with relay node. Journal of Computer Science and Technology, 2020, 35(5): 1115–1126. [doi: 10.1007/s11390-020-9576-4]

6 Tian YH, Song Y, Xia F, *et al*. Improving Chinese word segmentation with wordhood memory networks. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020. 8274–8285. [doi: 10.18653/v1/2020.acl-main.734]

7 章登义, 胡思, 徐爱萍. 一种基于双向 LSTM 的联合学习的中文分词方法. 计算机应用研究, 2019, 36(10): 2920–2924.

[doi: 10.19734/j.issn.1001-3695.2018.03.0239]
8 Cai TT, Ma ZY, Zheng H, *et al*. NE-LP: Normalized entropy- and loss prediction-based sampling for active learning in Chinese word segmentation on EHRs. Neural Computing and Applications, 2021, 33(19): 12535–12549. [doi: 10.1007/s00521-021-05896-w]
9 Devlin J, Chang MW, Lee K, *et al*. BERT: Pre-training of deep bidirectional Transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. Minneapolis: ACL, 2019. 4171–4186.
10 Yang ZL, Dai ZH, Yang YM, *et al*. XLNet: Generalized autoregressive pretraining for language understanding. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Vancouver: ACM, 2019. 517.
11 Diao SZ, Bai JX, Song Y, *et al*. ZEN: Pre-training Chinese text encoder enhanced by n-gram representations. Findings of the Association for Computational Linguistics: EMNLP. ACL, 2020. 4729–4740. [doi: 10.18653/v1/2020.findings-emnlp.425]
12 Tian YH, Song Y, Ao X, *et al*. Joint Chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020. 8286–8296. [doi: 10.18653/v1/2020.acl-main.735]
13 韩虎, 吴渊航, 秦晓雅. 面向方面级情感分析的交互图注意力网络模型. 电子与信息学报, 2021, 43(11): 3282–3290. [doi: 10.11999/JEIT210036]
14 Chen GM, Tian YH, Song Y, *et al*. Relation extraction with type-aware map memories of word dependencies. Findings of the Association for Computational Linguistics. ACL, 2021. 2501–2512.
15 Zhang ZS, Wu YW, Zhou JR, *et al*. SG-Net: Syntax-guided machine reading comprehension. Proceedings of the 2020 AAAI Conference on Artificial Intelligence. New York: AAAI, 2020. 9636–9643. [doi: 10.1609/aaai.v34i05.6511]
16 Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans: ACL, 2018. 464–468. [doi: 10.18653/v1/N18-2074]
17 Wang X, Tu ZP, Wang LY, *et al*. Self-attention with structural position representations. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong: ACL, 2019. 1403–1409. [doi: 10.18653/v1/D19-1145]
18 Nguyen DV, Vo LB, van Thin D, *et al*. Span labeling approach for Vietnamese and Chinese word segmentation. Proceedings of the 18th Pacific Rim International Conference on Artificial Intelligence. Hanoi: Springer, 2021. 244–258. [doi: 10.1007/978-3-030-89363-7_19]
19 Qiu XP, Pei HZ, Yan H, *et al*. A concise model for multi-criteria Chinese word segmentation with Transformer encoder. Findings of the Association for Computational Linguistics: EMNLP 2020. ACL, 2020. 2887–2897.
20 Chen XC, Shi Z, Qiu XP, *et al*. Adversarial multi-criteria learning for Chinese word segmentation. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver: ACL, 2017. 1193–1203. [doi: 10.18653/v1/P17-1110]
21 Gong JJ, Chen XC, Gui T, *et al*. Switch-LSTMs for multi-criteria Chinese word segmentation. Proceedings of the 2019 AAAI Conference on Artificial Intelligence. Honolulu: AAAI, 2019. 6457–6464. [doi: 10.1609/aaai.v33i01.33016457]
22 周裕林, 陈艳平, 黄瑞章, 等. 一种采用机器阅读理解模型的中文分词方法. 西安交通大学学报, 2022, 56(8): 95–103. [doi: 10.7652/xjtuxb202208010]
23 Maimaiti M, Liu Y, Zheng YH, *et al*. Segment, mask, and predict: Augmenting Chinese word segmentation with self-supervision. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2021. 2068–2077. [doi: 10.18653/v1/2021.emnlp-main.158]
24 Huang WP, Cheng XY, Chen LK, *et al*. Towards fast and accurate neural Chinese word segmentation with multi-criteria learning. Proceedings of the 28th International Conference on Computational Linguistics. Barcelona: ACL, 2020. 2062–2072. [doi: 10.18653/v1/2020.coling-main.186]
25 韩士洋, 马致远, 杨芳艳, 等. 针对中文分词的带标签注意力的成词记忆网络. 计算机应用研究, 2022, 39(6): 1651–1655. [doi: 10.19734/j.issn.1001-3695.2021.11.0592]

(校对责编: 牛欣悦)